

# Sunfa Ata Zuyan machine learning models for moon phase detection: algorithm, prototype and performance comparison

Ata Jahangir Moshayedi<sup>1</sup>, Zu-yan Chen<sup>1</sup>, Liefia Liao<sup>1</sup>, Shuai Li<sup>2</sup>

<sup>1</sup> School of Information Engineering, Jiangxi University of Science and Technology, Jiangxi, China

<sup>2</sup> College of Engineering, Swansea University, SA17EN, Swansea, UK

## Article Info

### Article history:

Received Sep 14, 2021

Revised Dec 17, 2021

Accepted Dec 28, 2021

### Keywords:

Edge detection

Embedded system

Moon date declaration

Moon phases

Morphology

Raspberry Pi

Sunfa Ata Zuyan

## ABSTRACT

The history recorded moon as the most inspiring object in the sky, but it combined with visibility issues to study the phases. This research paper proposes a novel algorithm named Sunfa Ata Zuyan (SAZ), which is meant to extend the shape detection algorithms to aim for lunar phase deceleration and overcome the difficulties encountered by the previous methods to find the moon and determine its phase. The paper sets to investigate two aims. First, propose the add-on algorithm SAZ to determine the lunar phase's data faster. Secondly, evaluate the Raspberry Pi as the main CPU due to its compact size and power as the primary processor based on the idea of a portable designed system. Then to examine the ability of the SAZ algorithm, it's combined with famous algorithms like hue, saturation and value (HSV), Canny, erosion, shape detection, and binarization has been tested on both personal computers (PC) and Raspberry Pi with the same images being compared. The results show that SAZ will help the shape detection algorithm to find the object and disclose the moon phases. Furthermore, the Raspberry Pi, functioning as a CPU, can perform as a hand-to-hand system to determine the lunar phase as a compact portable remote sensing structure.

This is an open access article under the [CC BY-SA](#) license.



## Corresponding Author:

Ata Jahangir Moshayedi

School of Information Engineering, Jiangxi University of Science and Technology

No 86, Hongqi Ave, Ganzhou, Jiangxi, 341000, China

Email: [ajm@jxust.edu.cn](mailto:ajm@jxust.edu.cn)

## 1. INTRODUCTION

From the early beginning, the moon has been interested. People have tried to observe it and collect information about its modes and shape, even regarded as a God worthy of worship. This attraction is probably because of its effect on the environment and the human body [1], [2]. The moon has more secrets to discover than just its dark side. It affects the earth's environment and creatures, like stock [3], weather, and bird migration [4]. Some recent papers even show physical and psychological effects on the human body, such as sleep, behaviour [5], and health side effects like renal colic [4], [6]. There are also important applications such as navigation at sea. This application can be used to study the moon as well [7]. One of the main study applications regarding the moon's declaration date is in the lunar calendar, which some countries follow and even align their religions and ethics with [7]. The moon, based on its position relative to the earth and sun, has different shapes and modes, which we call phases. These phases are named specifically to help people to recognize and identify them. As shown in Figure 1, these eight phases start from the closest place in the new moon phase, which achieves less visibility because it is located behind the earth. These lunar phases are named: i) new moon defined as the case that sun and moon are arrayed, and the sun and earth are located on opposite sides of the moon; ii) waxing crescent: that when moon appears to be partly, but less than one-half, illuminated by direct sunlight; iii) first quarter (half moon), moon's upper hemisphere come out to

be illuminated by direct sunlight; iv) waxing gibbous when moon phase appears to be more than one-half but not fully brightened by direct sunlight; v) full moon as the case that moon's lit opposite side of the earth; vi) third quarter (last quarter) that one-half of the moon come up to be illuminated by direct sunlight; vii) waning gibbous moon appear more than half, but it is not fully illuminated by direct sunlight; and viii) waning crescent the last phase starts just after the third quarter moon and continue until the following new moon phase [7].

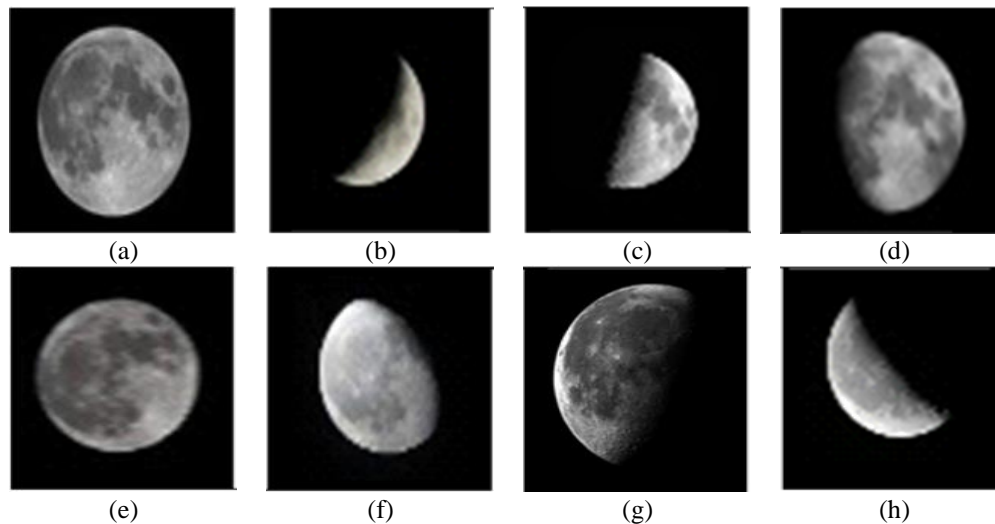


Figure 1. Eight phases of the moon: (a) new moon, (b) waxing crescent, (c) first quarter (halfmoon), (d) waxing gibbous, (e) full moon, (f) third quarter (last quarter), (g) waning gibbous, and (h) waning crescent

As the name suggests, each phase has a disparate appearance and explanation. There are some affinities between similarities in the last day and middle day of the month. Some problems, like cloudy nights, different times of visibility and percentage of illumination, make shape tracking difficult. Those days are called skipped days [7]. Some mobile applications (APPS) can observe the moon, but none can capture a picture of the moon in real-time. The moons' shape record for all cities is not available on the internet, which may cause problems in recognizing its shape and detecting [8] a lunar phase [9]. Because of the lack of stable systems to recognize the moon and its phases, researchers and designers' accurate investigation is required. On the other hand, having an effective algorithm concerning several aspects (number of code lines, accuracy, and response) to find and determine the lunar dates against various backgrounds in the form of a portable system is a big challenge. New developments in the embedded system allow the researcher to test a new system with features like small size, fast speed and low cost (such as Raspberry Pi) [10]. Our previous research paper investigated these problems, including portable systems with the Sunfa Ata Zuyan (SAZ) algorithm. The system was designed to capture and analyze the real moon pictures and estimate the moon date [11].

In this paper, along with the SAZ algorithm proposition, the comparison of the proposed algorithm with known algorithms is considered, and their performance on a personal computer (PC) and Raspberry Pi boards are summarized. This paper is arranged as follows; short reviews elaborate and compare different algorithms, including the proposed algorithm in section 2. The previously designed hardware system and components are described in section 3. In section 4 makes the comparison of six algorithms finding the moon in the picture, and their date declaration results in both a PC and Raspberry Pi. Finally, conclusions are presented with the achieved results and suggestions for future work. Based on the author's knowledge, the comparison between the Raspberry Pi and PC over the different algorithms has not been reported before that can be counted as the paper novelty.

## 2. RESEARCH METHOD

Images have many applications in recent research activities, industry, and human lives based on hidden data. This data has caused the proposal and development of different algorithms. In addition, the main

purpose of image processing is to give meaning to the different variations in picture intensity [12]. Based on this research's intended goal, the well-known algorithms for edge detection have been used to determine the edge plus line and morphological method. It uses the local pixel transformation to process the region shapes which are selected. The motive for selecting this method is the reputable research work that introduces it as a reliable method for finding a specific image in the picture. The edge detection method mainly attempts to reduce the image data with the intention of data minimization. The pixel boundary can distinguish two boundaries in the image [13], [14]. In other words, edge detection reduces the data size and uses only the necessary data among all the data there is. In the photo, edge detectors work based on identification and location. In this process, sharp, discontinuous sections of the image, which caused the image intensity and object changes, are detected [15], [16]. The morphological method works based on the shapes and is mainly used for binary images. This method works on the judgment of pixel zones with a pattern and focuses on the features like shape or morphology through logical transformations. In this way, morphology uses a structuring element that identifies small shapes and patterns. The benefits of mixing between edge detection and morphological method are reported [17], [18]. According to the previous statements, another target for this research was comparing, investigating, and proposing a new SAZ algorithm to detect and determine the date. The algorithms used are the Canny technique (the edge detection method), binarization, hue, saturation and value (HSV), erosion, and shape detection represent the morphological method [19].

In brief, HSV is a well-known algorithm that can detect and localize the target in pictures even with a low-cost camera presented with similar colours and robust to noises [20]. The input image from red, green, and blue (RGB) space is initially converted to HSV space into separate objects with different colours to use HSV. Then by using the medium filter, make the image clear. Afterwards, using the density-based spatial clustering of applications with noise (DBSCAN), the existing elements in the picture are classified, and the biggest one with the round outline is chosen as the target. In the end, the part of the image matrix which includes the target is added [21], [22]. Erosion (minkowsky subtraction) works based on two processes of eliminating and shrinking boundary points and is mainly used for binary images but can also be applied to grayscale images [12], [19], [22], [23]. Erosion mainly operates on the white part (highlighted part) and finds the highlighted parts of the original image, which are corroded, and the neighbourhood is removed from the image [24]. Dilation (minkowsky addition) [12] is mainly used for binary images but can be extended for grayscale images. It typically engages a structuring part for searching and increasing the shapes contained in the input photo image. In this method, the zones of foreground pixels get enlarged. Then the areas of foreground get maximized, and the background holes get minimized. Dilation also occurs for the white part (highlighted part) and expands the image's highlighted part (neighbourhood expansion). The affected image also has a larger highlighted area than the original image [17], [24].

Canny is a famous edge detection algorithm created in 1986 [25], [26]. Although some researchers know it as low-level processing [25], [26], it can detect the shape successfully used on edge detection [12]. This procedure works based on the image gradient calculation and uses four gradient operators to calculate different standards. In this algorithm, first, the Gaussian filter is used to smooth the image and filter noise. Then, after that, non-maximum suppression has been applied to eliminate stray responses caused by edge detection. As the next step, double-threshold detection is applied to determine the real and potential edges. Finally, the edge has been detected by deterring the isolated weak edge [27]. Binarization is used for binary images, the basis of segmentation [28], [29]. In this technique, initially, the image is changed into grayscale with the threshold value. In this method, the grey image (0)-(256) grey levels changes into black and white image (0 or 1) [29], [30] and remove lightness (brightness, density) as a feature of images. binarization is in common use in picture processing and leads to a good result. But the problem is how to choose a better threshold of binarization. It is always used with another handling method because it is weak in dealing with complex surroundings [29].

Shape detection is one of the blob detection methods based on shape [30], [31]. In this algorithm, the bilateral filtering on each pixel gives a smooth effect on the image. The average of the neighbourhood pixels gives the filter in each pixel, and then the edge detection with the help of the Canny algorithm is implemented. Then the contours find a base and connect the points, creating a similar intensity where colour is considered. Next, the act of finding circular shapes with the help of approximates a curve or a polygon (APPROXPOLYDP) will occur. The APPROXPOLYDP support finds the polygon shape by counting the polygon's number of vertices, known as the circular shape with a higher number of points [32]. This algorithm works by using OpenCV API and can detect the shape of objects in the photo. For better accuracy, need to specify the target to classify the detected objects [8], [14], [33], [34].

SAZ Novel algorithm the main target of this research is to have a simple algorithm that can be implemented with python on the Raspberry Pi platform. The proposed algorithm SAZ aimed to separate the target from the image by colour signature. Figure 2 shows the steps, and algorithm 1 gives the pseudocode for the SAZ. As it showed in this algorithm, some initial stages are similar to the HSV method, in the first

four steps, the algorithm to make an effective way for the moon phase and date declarations and introduced the technique to separate the specific targets from the picture.

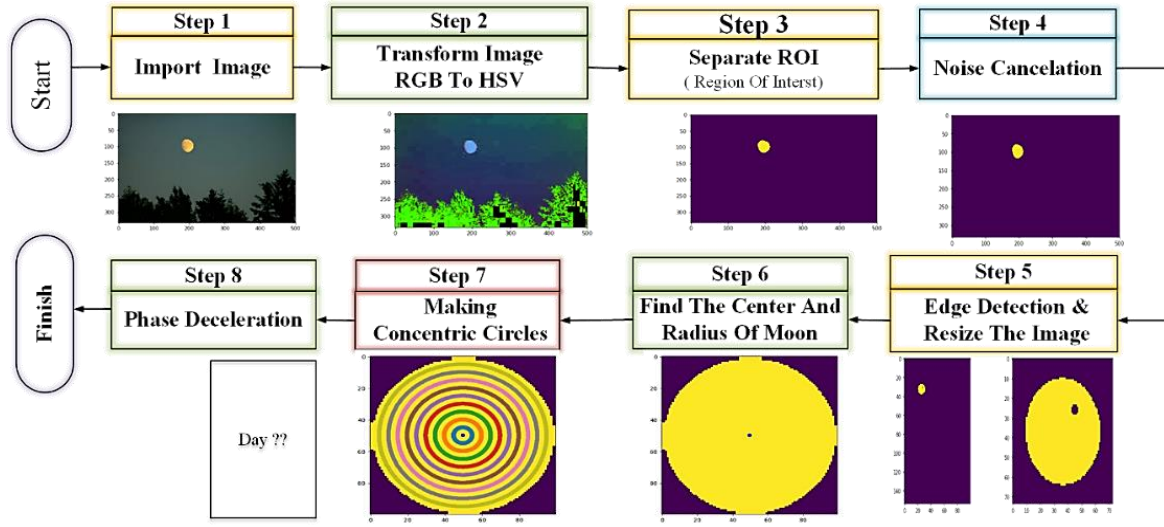


Figure 2. The SAZ Algorithm steps and brief

#### Algorithm 1. SAZ

Input: Import the color image array  $x \leftarrow (R, G, B)$

1. Transform image from RGB space to HSV space:

$$v = \max(R, G, B)$$

$$s = \begin{cases} v - \frac{\min(R, G, B)}{v} & \text{if } v \text{ is not } 0 \\ 0 & \text{Other wise} \end{cases}$$

$$h = \begin{cases} \frac{60 * (G - B)}{v - \min(R, G, B)} & \text{if } v = R \\ 120 + \frac{60 * (B - R)}{v - \min(R, G, B)} & \text{if } v = G \\ 240 + \frac{60 * (R - G)}{v - \min(R, G, B)} & \text{if } v = B \end{cases}$$

$$h = \begin{cases} h & \text{if } h \geq 0 \\ h + 360 & \text{if } h < 0 \end{cases}$$

$$\text{Output} \leftarrow (h, s, v)$$

2. Separate region of interest (ROI):  $\text{output } i, j = \begin{cases} 1; & 200 > \text{output } i, j, h > 0 \\ & 80 > \text{output } i, j, s > 0 \\ & 255 > \text{output } i, j, v > 140 \\ 0; & \text{Other wise} \end{cases}$

3. Noise cancellation:  $g(i, j) = \frac{1}{M} \sum_{\text{output} \in S} \text{output } i, j$

4. Edge detect and resize the image: DBSCAN utilized for object detection referring to energy density  
 $g(i, j)^2: g'(i, j) \leftarrow \text{DBSCAN}(g(i, j))$

5. Edge detection for shape:  $g''(i, j) \leftarrow \text{Edge}(g'(i, j))$

Find the centre and radius of moon *moon centre finding by unique point (A, B, C) on the circle:*

$$6. \quad \begin{cases} x = \frac{-(Ay - By)(Bx + By - Cx - Cy) + (By - Cy)(Ax + Ay - Bx - By)}{(Ax - Bx)(By - Cy) - (Ay - By)(Bx - Cx)} \\ y = \frac{(Ax - Bx)(Bx + By - Cx - Cy) + (By - Cy)(Ax + Ay - Bx - By)}{(Ax - Bx)(By - Cy) - (Ay - By)(Bx - Cx)} \end{cases}$$

Concentric circles to demarcate Moon:  $S'_{i,j} \leftarrow \text{Circles}(g''(i, j))$

7. Phase deceleration calculate the difference value:  $\text{difference} = (\sum_{i=0}^1 \sum_{j=0}^{10} S_{i,j} - S'_{i,j}) / 22$   
Output: Moon date declaration:  $\text{date} = \text{argmin}(\text{difference})$

As Figure 2 and algorithm1 show, SAZ has seven steps. Like the HSV in SAZ, it also needs to change the picture from RGB space to HSV space (image transform from RGB to HSV). With the aim of target object separation and region of interest (ROI) separation. The HSV format separates the different colour objects and then applies DBSCAN to lock the destination [7]. Then, the median filter makes the picture clear in the noise cancellation step. DBSCAN is used to remove large artefacts of noise in the picture, and the medium filter is used to omit some small artefacts of noise. In the next step (edge detection and image resizing using the concentric circles), SAZ uses the geometric method to find the centre of the target and draw 11 concentric circles on the target, divided into the left and right parts. After that, SAZ in the concentric circle's step calculates the percentage of every part and compares its value with the origin data. Finally, given the differences, SAZ will give the phase declaration [11]. This algorithm needs seven libraries named NumPy, pillow, scipy, matplotlib scikit, learn, and scikit-image to implement on python.

### 3. HARDWARE DESIGNED AND SYSTEM COMPONENT

To collect and analyze the moon phase and date declaration, the hardware system has been designed and has been planned to test based on the SAZ algorithms (Figure 3(a) and Figure 3(b)). Figure 3 shows the portable design hire the Raspberry Pi recognized as the smallest computer [10]. The designed system as it is shown (Figure 3(a) and Figure 3(b)) includes the Raspberry Pi board and camera, a 7-inch liquid crystal display (LCD) and graphical user interface (GUI) section (Figure 3(c)) to interact with the user. The designed system is elaborated in our previous research paper [11].

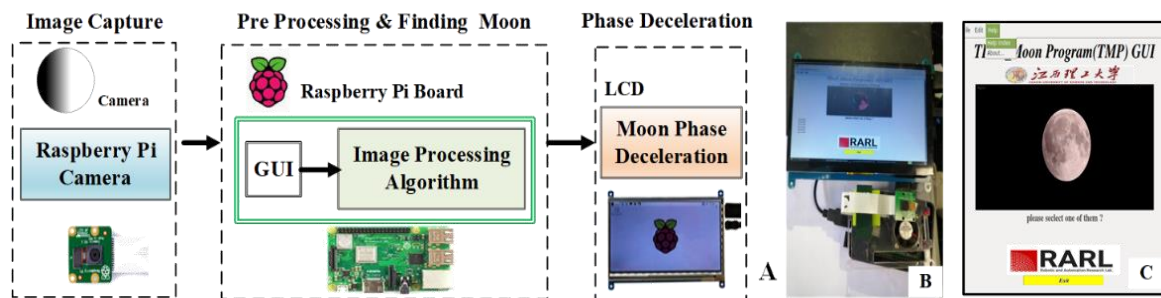


Figure 3. The designed hardware part (a) the system block, (b) design hardware, and (c) designed GUI [11]

### 4. ALGORITHMS EVALUATION RESULT AND COMPARISON

After the system has successfully captured the moon images, the next target is its performance comparison which has been done in two steps. First, finding the moon as the initial goal and then the date deceleration. In addition to all the pointed algorithms on Raspberry Pi, the difference between Raspberry Pi and a regular PC (Intel (R) -I 7- 6700HQ/ @2.60 GHz/ Windows 10/ Python IDE version 3 : Anaconda-Spyder IDLE) is investigated. As mentioned before, the test steps (finding the moon and date declaration phases) are verified with two aspects. First, experiments were carried out on a fixed sample taken from [9], and then, some real and random images of the moon have taken by the designed system. To compare both stages, the algorithms are named HSV, erosion, dilation, Canny, binarization [35], and shape detection, as mentioned before. The tested result is mentioned in this section.

#### 4.1. Moon finding on the fixed data set

The main aim of this section is to find the moon as the key target in the image. Besides, this part shows the algorithm accuracy and its ability to find the moon over the different objects. To do the test, the used fixed set of the picture [9] related to different phases of the moon used as is shown in Figure 4. To test fixed moon images, initially, the HSV algorithm and other algorithms named erosion, dilation, Canny, binarization and shape detection are tested, and the results are shown in Table 1 and Figure 5. The results illustrate the image detection in PC and Raspberry Pi, (using python language with two different versions of 3.6 in PC and 2.7 version in Raspberry Pi). The recent version and library improvements cause different python versions with different results after using the recent library [36]. The used libraries names are numpy, pillow, scipy, matplotlib, scikit-learn, cv2, scikit-image

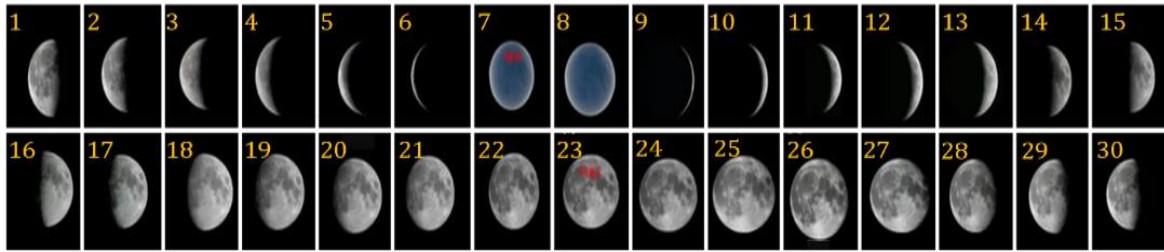
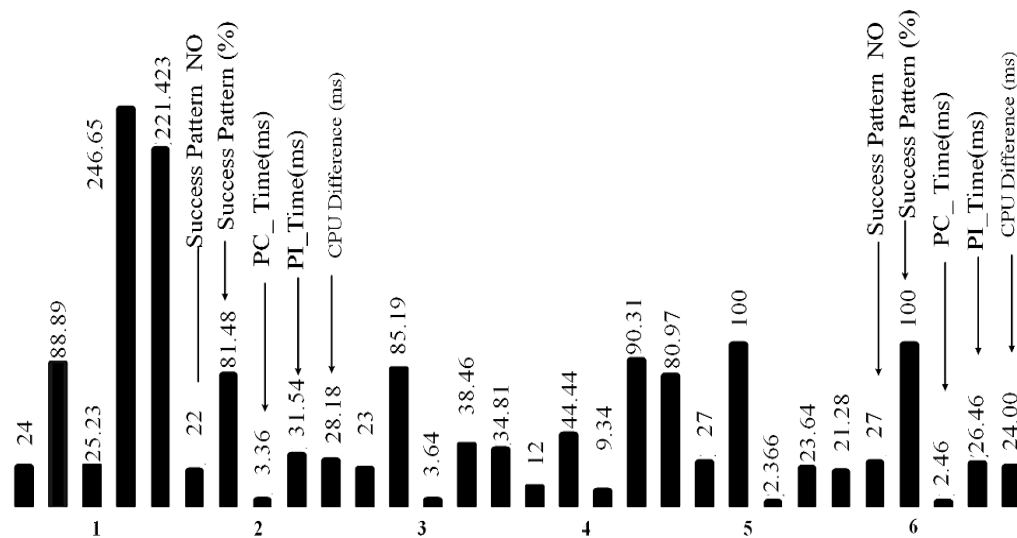


Figure 4. The fixed moon picture sample [9]

Table 1. PC and Raspberry Pi performance for Moon finding over the 27 fixed sample images

Method	Number of Success pattern	Success Pattern (%)	PC Average time(ms)	PI Average time(ms)	CPU Difference (ms)	file size (KB)	Number Of Code Lines	Method Difficulty
HSV	24	88.89	25.23549	246.658423	221.423	5.5	143	No
Erosion	22	81.48	3.365941	31.549463	28.1835	5.5	144	
Dilation	23	85.19	3.6456841	38.462884	34.8172	5.4	141	
Canny	12	44.44	9.345688	90.3165486	80.97086	5.5	143	Yes
Binarization	27	100	2.366168	23.6468	21.2806	5.4	142	
Shape detection	27	100	2.4618135	26.46548	24.0037	5.7	147	

Figure 5. The various algorithm performance over the fixed moon images  
1. HSV, 2. Erosion, 3. Dilation, 4. Canny, 5. Binarization, and 6. Shape detection

As it dissipated in Table 1 and Figure 3, the algorithm performance in Raspberry Pi takes more time than a personal computer. Comparing the algorithms to find the Moon with the fixed moon phase pattern shows that shape detection and binarization have 100% success (maximum). Canny has the minimum success rate for finding the Moon in the picture. HSV has the maximum time taken over the processor platforms. To compare the difficulty from different points of view, the result shows that the HSV, erosion and dilation have the least difficulty compared to others. The lowest number of used libraries in programming code belongs to HSV, and the maximum one belongs to shape detection. The number of lines in the dilation code is the least, and shape detection has the maximum number of code lines. This point is common in all algorithms, so the comparison is not repeated for all steps. The same practice is repeated for the real moon photos with different sizes, and the obtained results are shown in Table 2. From the eight sampled photos Figure 6, six are taken by hardware, and two are taken from another source regarding the noisy background. Table 2, Figure 6 and Figure 7 show the algorithm performance based on PC and Raspberry Pi. Figure 7 shows the real outcome of



different algorithms named HSV, erosion, dilation, Canny, binarization, and shape detection algorithms process on captured moon image on the eight-sample image. As mentioned before, among the used eight sampled photos, six are taken by hardware, and two are taken from another source regarding the noisy background

Table 2. Real Moon detection in picture over Total sample 8 samples

Method	Success Pattern (%)	PC Average time(ms)	PI Average time(ms)	CPU Difference (ms)
HSV	62.5	986.313	1616.849	630.535
Erosion	25	186.645	933.548	746.904
Dilation	37.5	354.684	1464.851	1110.167
Canny	25	17.647	863.149	845.502
Binarization	50	9.315	56.316	47.001
Shape detection	100	13.543	676.465	662.922

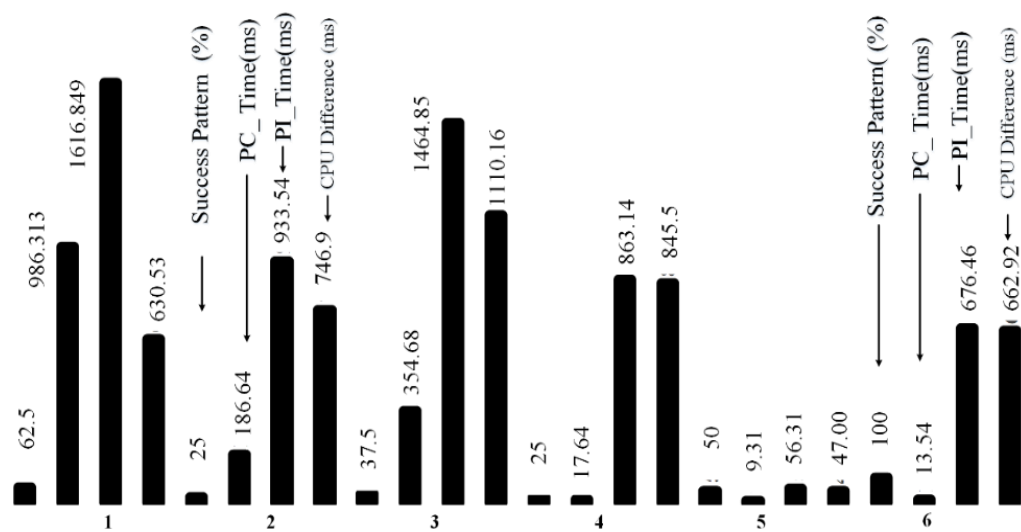


Figure 6. Finding the moon over the real sample, CPU (PC and PI) with success rate performance 1. HSV, 2. Erosion, 3. Dilation, 4. Canny, 5. Binarization, and 6. Shape detection

Based on Figure 6 and Table 2, and as shown in Figure 7, the maximum number of success pattern for finding the moon belongs to the shape detection, and HSV has the second grade. Furthermore, the minimum average time belongs to binarization and the maximum time is the HSV algorithm. Besides, Figure 7 shows that shape detection successfully detected the moon in the image over the other tested algorithms.

#### 4.2. Phase declaration test

The main problem for the mentioned algorithm, over finding the proper object in the image, is their performance rather than focusing on the target. Then, in the second stage, to show the SAZ as the proposed algorithm, the mentioned algorithm is mixed with SAZ to give a better guideline for image detection. In other words, the SAZ helps the algorithms to decelerate the phase, and the main algorithm tries to find the moon in the image. The algorithm performance with the fixed 27 samples results is shown in Table 3 and Figure 8.

As the result shows, with the help of SAZ, all 27 samples could detect and declare the moon phase by the Shape detection method. The obtained results show SAZ and Shape detection take 0.646 minutes on PC and 6.657 minutes by using the Raspberry Pi to process the 27-sample image. Even the less Average time belongs to the Canny algorithm, but the percentage of success rate is least between other tested algorithms.

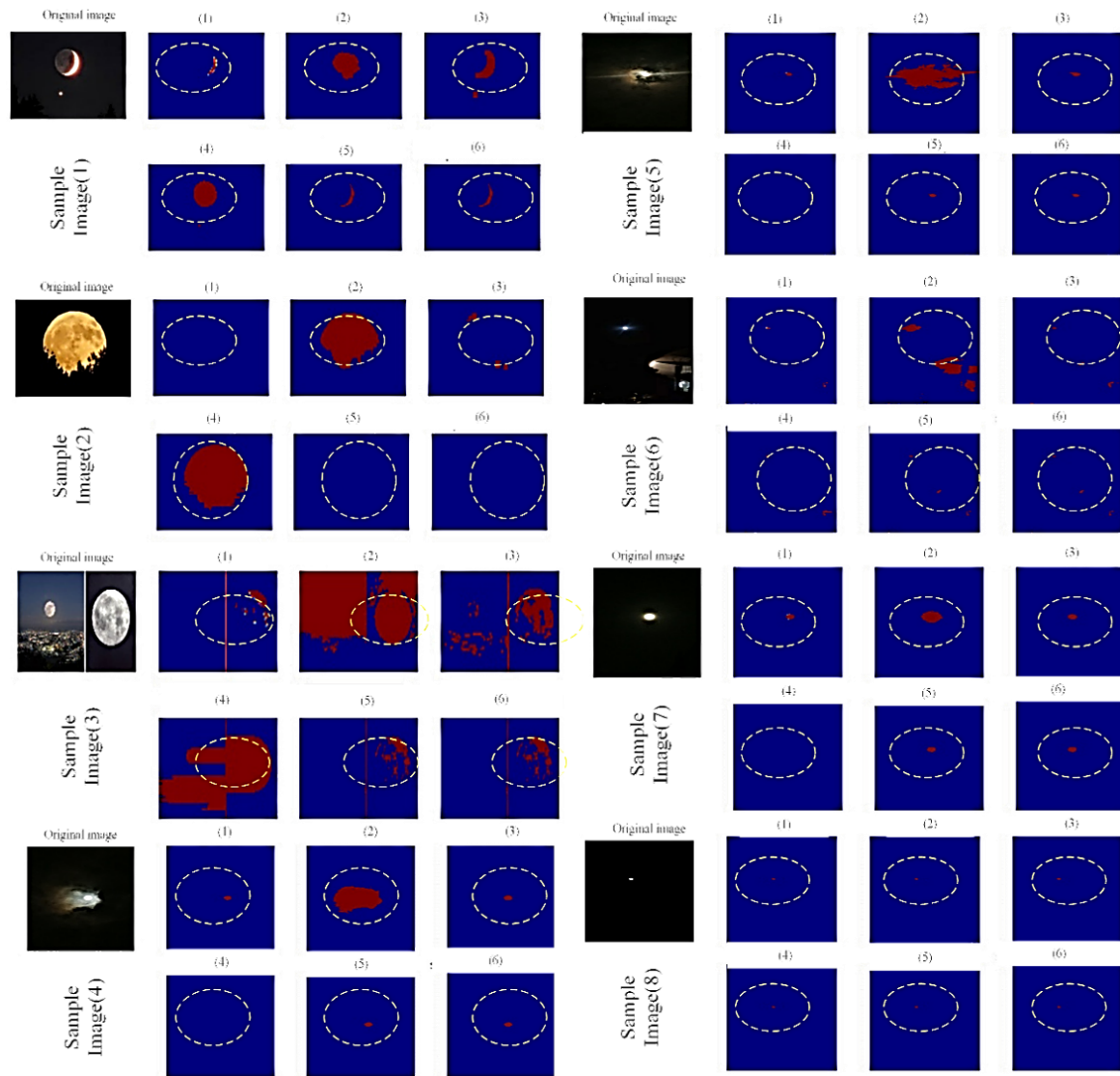


Figure 7. The eight sample photos, five have been taken by designed hardware, and three images have been selected at random: 1.HSV , 2. Erosion, 3. Dilation, 4. Canny, 5. Binarization, 6. Shape detection

Table 3. The SAZ and various algorithm performance combinations for moon phase deceleration

Method	HSV	Erosion	Dilation	Canny	Binarization	Shape detection
PC average time(ms)	0.632	0.685	0.647	0.652	0.643	0.646
PI average time(ms)	6.431	7.034	6.613	6.713	6.602	6.657
CPU difference(ms)	5.8	6.35	5.97	6.06	5.96	6.01
Number of success pattern	14	15	14	7	14	27
Success pattern (%)	51.85	55.56	51.85	25.93	51.85	100
File size (KB)	5.5	5.5	5.4	5.5	5.4	5.7
Number of code line	143	144	141	143	142	147

#### 4.3. Phase declaration on real moon photo

The previous results show that shape detection with SAZ displays a 100 percent success rate in finding and detecting the moon phase mode. Then, to better analyses SAZ, the algorithm performances are tested on an eight-moon sample. These samples are mixed with a noisy background, and they are the real moon images. Furthermore, to compare the result, mean-square error (MSE) and peak signal to noise ratio (PSNR) with the processor performance are listed in Table 4 and are shown in Figure 9.



As shown in Table 4 and Figure 9, all eight samples were successfully detected. The lunar phase is based on SAZ proposed algorithm ability, and the minimum processor time belongs to sample 5. The current research can be hired on other applications like determining people's faces, inside pipeline defects [36], odor shape in odor tracking systems [37], extracting faces and palms from images [38] and other detection and analysis applications [39]-[42].

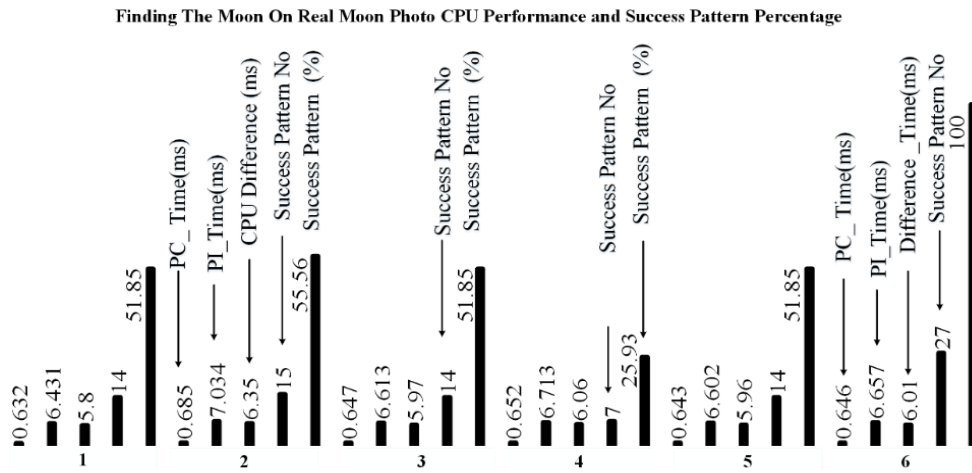


Figure 8. SAZ and various algorithm performance on Phase deceleration, CPU and Success patterns percentage over the eight real moon images: 1. HSV, 2. Erosion, 3. Dilation, 4. Canny, 5. Binarization, and 6. Shape detection

Table 4. Phase declaration and various algorithm performance

Total sample = 27	1	2	3	4	5	6	7	8
Evaluation	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
PSNR	45.485	26.927	22.531	20.214	18.776	10.351	11.16	20.03
MSE	1.839	131.947	363.089	618.924	862.027	5998.36	4978.282	645.776
Determining lunar phase	17th	14th	24th	19th	15th	14th	11th	3th
AVG time on PC(Second)	0.019941	0.029926	0.019951	0.015972	0.009983	0.089777	0.039899	0.04987
AVG time on PI (Second)	0.179814	0.186056	0.180183	0.143276	0.054044	0.72887	0.62788	0.639181
CPU difference performance	0.159873	0.15613	0.160232	0.127304	0.044061	0.639093	0.587981	0.589311

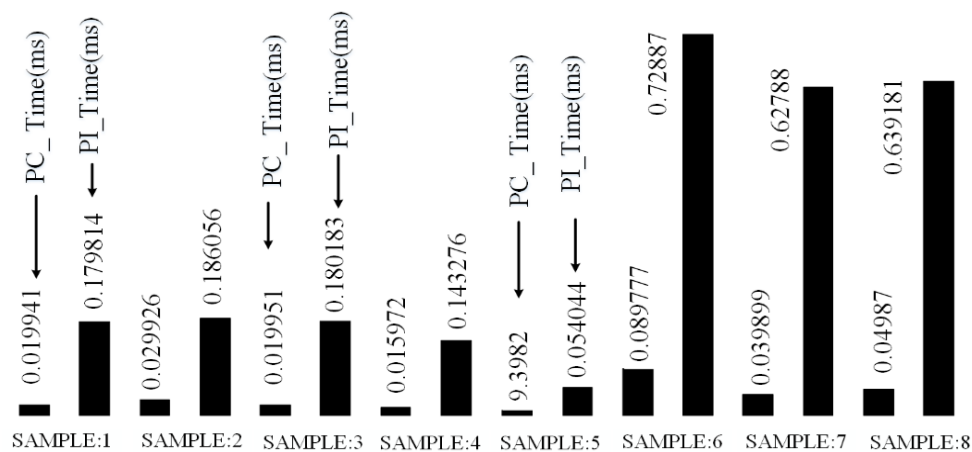


Figure 9. The date declaration and 8 real moon samples with the SAZ algorithm performance on PC and PI

## 5. CONCLUSION

Finding a lunar phase, both out of human interest and for calendars, has always been challenging. There are lots of other usages like navigation, which are explored. In this research, the moon phase

determination is based on designing a portable system to test the various algorithms we have considered. The paper sets to investigate two aims. First, propose the add-on algorithm SAZ to determine the lunar phase's data faster. Secondly, test and compare the Raspberry Pi as the main CPU due to its small size and ability as the main processor based on the idea of the portable designed system. In this comparison, HSV, Canny, erosion, shape detection and binarization algorithm as the famous ones tested on the Raspberry Pi and PC. The mentioned algorithms were selected based on a good performance in other reported work. The sample picture used to test the SAZ algorithm has been split into the first 27 sample photos due to 8 major phases of the moon. Then, eight numbers of moon samples are tested. Six were taken by the Raspberry Pi camera (designed system), and two were used to represent the noisy picture with a greater number of objects. The tested sample considers two stages of finding the moon in image and date determination. The result of the combination of SAZ and the other algorithms shows that the HSV algorithm can resist noise and can be easily migrated to another device. Compared with others, it uses fewer python libraries and doesn't need other settings (such as thresholds or other things). But its weakness is that it takes more time than other algorithms. The dilation and erosion algorithm have a fair skill to resist noise and can be easily realized. But it has demerits. The larger the objects, the lower the accuracy. The main problem for this algorithm reported before is its difficulty finding the optimal binarization. It means that a different algorithm shows a different performance.

The Canny algorithm was more effective in other papers (when the dual-threshold is good for the target), but this paper's result was not the same. The Canny algorithm is effective with a smooth image, but it is difficult to realize. Besides, its dual-threshold is also hard to ensure. Shape detection can find all objects in the picture, but it needs to specify the target. It has the advantage of finding everything in the image, but the drawback is that it's hard to ensure the explored object is a target or not. Then, the SAZ as the proposed algorithm can help to detect the targets and navigate them to determine the date. The result shows that the HSV, erosion, and dilation have the least difficulty to compare an algorithm's difficult point of view. In python language, coding the least number of libraries goes to HSV, and the maximum number belongs to the shape detection. The number of code lines in dilation is the least, and shape detection has the maximum number of the code line. The results obtained indicate that the processing time is lengthy compared to PC, but the Raspberry Pi has to handle the lunar phase determination job. In conclusion, the result evaluation shows, even if the number of real samples is limited, which needs using the focus lens of the Raspberry Pi camera. However, the total SAZ performance shows its ability to disclose the moon phase by extending the shape detection algorithm with a better result as an additional algorithm. The present study could be used on various research topics such as face and palms recognition, pipeline defects detection, odor types tracking systems and other detection and analysis applications.

## ACKNOWLEDGEMENTS

This work was supported by Jiangxi University of Science and Technology, 341000, Ganzhou, P.R China, underfunding number: 2021205200100563.

## REFERENCES





- [1] N. Yousfi, M. A. Mejri, M. Rouissi, A. Hammami, M. Tabben, A. Chaouachi, M. Haddad, and K. Chamari, "Effects of lunar phases on short-term, explosive physical performance among young trained athletes," *The Journal of Biological and Medical Rhythm Research*, vol. 35, Issue 4, pp. 565–572, Jan. 2018, doi:10.1080/07420528.2017.1422741.
- [2] M. Mishra, "Traditional Knowledge Systems, Culture and Environmental Sustainability: Concepts from Odisha, India," In: Prasad K. (eds) "Communication, Culture and Ecology. Communication, Culture and Change in Asia", vol. 6. Springer, Singapore, Nov. 2017, doi: 10.1007/978-981-10-7104-1\_4.
- [3] I. D. Dichev and T. D. Janes, "Lunar cycle effects in stock returns," *The Journal of Private Equity*, vol. 6, no. 4, pp. 8–29, Aug. 2003, doi: 10.3905/jpe.2003.320053.
- [4] P. Pyle, N. Nur, R. P. Henderson, and D. F. DeSante, "The Effects of Weather and Lunar Cycle on Nocturnal Migration of Landbirds at Southeast Farallon Island, California", *The Condor*, vol. 95, no. 2, pp. 343–361, May 1993, doi:10.2307/1369357.
- [5] C. Cajochen, S. A-Ekici, M. Münch, S. Frey, V. Knoblauch, and A. W-Justice, "Evidence that the lunar cycle influences human sleep," *Current Biology*, vol. 23, no. 15, pp. 1485–1488, Aug. 2013, doi: 10.1016/j.cub.2013.06.029.
- [6] H. Tokgöz, S. Yalçınkaya, E. İslamoğlu, K. Karamık, Ö. Tokgöz, and M. Savaş, "Lunar cycle may have an effect on Shock Wave Lithotripsy related pain outcome," *Ghana Medical Journal*, vol. 51, no. 4, pp. 181–186, Dec. 2017, doi:10.4314/gmj.v51i4.6.
- [7] M. T. N-Garcia and A. A. R-Chelén, "More than noise: light, moon phase, and singing behavior in a passerine," *Urban Ecosystems*, Aug. 2021, doi:10.1007/s11252-021-01142-2.
- [8] M. N. Abdulwahed and A. K. Ahmed, "Improved anti-noise attack ability of image encryption algorithm using de-noising technique," *Telkomnika*, vol. 18, no. 6, pp. 3080–3087, Jun. 2020. doi: 10.12928/telkomnika.v18i6.16384.
- [9] C. Witt, "Phases and full moon names," B&H Explora, 17-Jan-2019. Accessed: 07-Dec-2021 [Online]. Available: <https://www.bhphotovideo.com/explora/outdoors/features/phases-and-full-moon-names>

- [10] A. J. Moshayedi, A. S. Roy, L. Liao, and S. Li, "Raspberry Pi SCADA Zonal based System for Agricultural Plant Monitoring," *2019 6th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 427-433, Jun. 2020, doi: 10.1109/ICISCE48695.2019.00092.
- [11] A. J. Moshayedi, Z. Chen, L. Liao, and S. Li, "Portable Image Based Moon Date Detection and Declaration: System and Algorithm Code Sign," in *2019 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 1-6, Jun. 2019, doi: 10.1109/CIVEMSA45640.2019.9071604.
- [12] R. M. Yousaf, H. A. Habib, H. Dawood, and S. Shafiq, "A Comparative Study of Various Edge Detection Methods," *2018 14th International Conference on Computational Intelligence and Security (CIS)*, pp. 96-99, Nov. 2018, doi: 10.1109/CIS2018.2018.00029.
- [13] S. K. Roy, R. Mondal, M. E. Paoletti, J. M. Haut, and A. Plaza, "Morphological Convolutional Neural Networks for Hyperspectral Image Classification," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 8689-8702, June 2021, doi: 10.1109/JSTARS.2021.3088228.
- [14] I. K. Maitra, S. Nag, S. K. Bandyopadhyay, and T. Kim, "A Novel Approach to Detect Accurate Breast Boundary in Digital Mammogram Using Binary Homogeneity Enhancement Algorithm," *2011 International Conference on Ubiquitous Computing and Multimedia Applications*, pp. 71-75, Apr. 2011, doi: 10.1109/UCMA.2011.24.
- [15] Y. Wei and W. Badawy, "A new moving object contour detection approach," *IEEE International Workshop on Computer Architectures for Machine Perception*, May 2003, doi: 10.1109/CAMP.2003.1598168.
- [16] P. Ganesan and G. Sajiv, "A comprehensive study of edge detection for image processing applications," *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pp. 1-6, Mar. 2017, doi: 10.1109/ICIIECS.2017.8275968.
- [17] B. Chan, H. Lim, L. Gopal, A. Gopalai, and D. Gouwanda, "Stereo vision human motion detection and tracking in uncontrolled environment," *Telecommunication Comput. Electron. Control (Telkomnika)*, vol. 16, no. 3, pp. 955-964, Jun. 2018, doi: 10.12928/telkomnika.v16i3.9033.
- [18] A. N. Evans and X. U. Liu, "A morphological gradient approach to color edge detection," in *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1454-1463, Jun. 2006, doi: 10.1109/TIP.2005.864164.
- [19] Y. A. Akter, M. A. Rahman, and M. O. Rahman, "Quantitative analysis of Mouza map image to estimate land area using zooming and Canny edge detection," *TELKOMNIKA*, vol. 18, no. 6, pp. 3293-3302, Jul. 2020, doi: 10.12928/telkomnika.v18i6.16179.
- [20] A. Aqthobillobbany, A. N. Handayani, D. Lestari, Muladi, R. A. Asmara and O. Fukuda, "HSV Based Robot Boat Navigation System," *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, pp. 269-273, Nov. 2020, doi: 10.1109/CENIM51130.2020.9297915.
- [21] P. Assiroj, H. Warnars, E. Abdurrachman, A. I. Kistijantoro, and A. Doucet, "Measuring memetic algorithm performance on image fingerprints dataset," *Telkomnika*, vol. 19, no. 1, pp. 96-104, Aug. 2020, doi: 10.12928/telkomnika.v19i1.16418.
- [22] T. Nguyen, A. Khosravi, A. Bhatti, D. Creighton, and S. Nahavandi, "Neural signal analysis by landmark-based spectral clustering with estimated number of clusters," *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 4042-4049, Jul. 2014, doi: 10.1109/IJCNN.2014.6889674.
- [23] Dongli Lv, Z. Lian, T. Zhang, "Study of Cavitation and Cavitation Erosion Quantitative Method Based on Image Processing Technique," *Advances in Civil Engineering*, Jul. 2018, doi: 10.1155/2018/5317578.
- [24] X.-Y. Gong, H. Su, D. Xu, Z.-T. Zhang, F. Shen, and H.-B. Yang, "An overview of contour detection approaches," *International Journal of Automation and Computing*, vol. 15, no. 6, pp. 656-672, Dec. 2018, doi: 10.1007/s11633-018-1117-z.
- [25] L. Ding and A. Goshtasby, "On the Canny edge detector," *Pattern Recognit*, vol. 34, no. 3, pp. 721-725, Mar. 2001, doi: 10.1016/S0031-3203(00)00023-6.
- [26] R. V. K. Reddy, K. P. Raju, M. J. Kumar, L. R. Kumar, P. R. Prakash, and S. S. Kumar, "Comparative Analysis of common Edge Detection Algorithms using Pre-processing Technique," *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 7, No. 5, pp. 2574-2580, Oct. 2017, doi: 10.11591/ijece.v7i1.pp2574-2580.
- [27] I. Mednieks, "Object detection in grayscale images based on covariance features," *2008 International Conference on Signals and Electronic Systems*, pp. 205-208, Sept. 2008, doi: 10.1109/ICSES.2008.4673393.
- [28] Y. Zhang, T. Gao, D. Li, and H. Lin, "An improved binarization algorithm of QR code image," *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pp. 2376-2379, Apr. 2012, doi: 10.1109/CECNet.2012.6201551.
- [29] X. Liu, B. Zhang, B. Liu, and Y. Lan, "Phong model based binarization and its application to billet images with non-uniform illumination," *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1763-1767, Jun. 2017, doi: 10.1109/ICIEA.2017.8283124.
- [30] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of Adjacent Contour Segments for Object Detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 36-51, Jan. 2008, doi: 10.1109/TPAMI.2007.1144.
- [31] H. Moon, R. Chellappa, and A. Rosenfeld, "Optimal edge-based shape detection," in *IEEE Transactions on Image Processing*, vol. 11, no. 11, pp. 1209-1227, Nov. 2002, doi: 10.1109/TIP.2002.800896.
- [32] R. Bansal, G. Raj, and T. Choudhury, "Blur image detection using Laplacian operator and Open-CV," *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, pp. 63-67, Nov. 2016, doi: 10.1109/SYSMART.2016.7894491.
- [33] S. A. Aziz et al., "A review on region of interest-based hybrid medical image compression algorithms," *Telecommunication Computing Electronics and Control (Telkomnika)*, vol. 18, no. 3, pp. 1650-1657, Jun. 2020, doi: 10.12928/TELKOMNIKA.v18i3.14900.
- [34] R. J. Meimban, A. R. Fernando, A. Monsura, J. Rañada, and J. C. Apduhan, "Blood Cells Counting using Python OpenCV," *14th IEEE International Conference on Signal Processing (ICSP)*, pp. 50-53, Aug. 2018, doi: 10.1109/ICSP.2018.8652384.
- [35] S. S. Elkapelli and L. B. Damahe, "A review: Region of interest based image retrieval," *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, pp. 1-6, Nov. 2016, doi: 10.1109/GET.2016.7916731.
- [36] A. J. Moshayedi, S. S. Fard, L. Liao, and S. A. Eftekhari, "Design and Development of Pipe Inspection Robot Meant for Resizable Pipe Lines," *International Journal of Robotics and Control*, vol. 2, no. 1, p. 25, Jan. 2019, doi: 10.5430/ijrc.v2n1p25.
- [37] A. J. Moshayedi and D. C. Gharpure, "Evaluation of bio inspired Mokhtar: Odor localization system," *18th International Carpathian Control Conference (ICCC)*, pp. 527-532, May. 2017, doi: 10.1109/CarpathianCC.2017.7970457.
- [38] A. J. Moshayedi, J. Li and L. Liao, "Simulation study and PID Tune of Automated Guided Vehicles (AGV)," *2021 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, Jun. 2021, pp. 1-7, doi: 10.1109/CIVEMSA52099.2021.9493679.





- [39] A. H-Najafabadi, S. Salmanpour, F. Sen, P. N. Asrami, M. Mahdavian, and M. A. Khalilzadeh, "A Tramadol Drug Electrochemical Sensor Amplified by Biosynthesized Au Nanoparticle Using Mentha aquatic Extract and Ionic Liquid," *Topics in Catalyst*, pp. 1-8, Sep. 2021, doi: 10.1007/s11244-021-01498-x.
- [40] A. H-Najafabadi, A. Ghasemi, and R. Mozaffarinia, "Synthesis and evaluation of microstructural and magnetic properties of Cr 3+ substitution barium hexaferrite nanoparticles (BaFe 10.5– x Al 1.5 Cr x O 19)," *Journal of Cluster Science*, vol. 27, no. 3, pp. 965–978, Jan. 2016, doi: 10.1007/s10876-015-0963-x.
- [41] S. A. Baker, H. H. Mohammed, and H. A. Aldabagh, "Improving face recognition by artificial neural network using principal component analysis," *Telecommunication Computing Electronics and Control (Telkomnika)*, vol. 18, no. 6, pp. 3357–3364, Jun. 2020, doi : 10.12928/telkomnika.v18i6.16335.
- [42] N. Ukani and H. Mehta, "An accurate method for license plate localization using morphological operations and edge processing," *2010 3rd International Congress on Image and Signal Processing, 2010*, pp. 2488-2491, Oct. 2010, doi: 10.1109/CISP.2010.5647926.

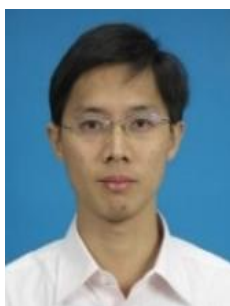
## BIOGRAPHIES OF AUTHORS







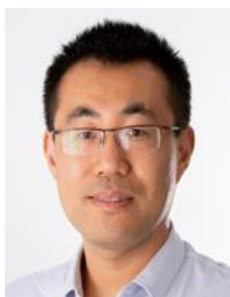
**Dr. Ata Jahangir Moshayedi**     Associate professor at Jiangxi University of Science and Technology, China, PhD. In Electronic Science from Savitribai Phule Pune University, India (2015), member of various Societies like IEEE, Instrument Society of India as a Life Member, Lifetime Member of Speed Society of India, and editorial team of various conference and journals. His research interest includes Robotics and Automation/Sensor modelling /Bio-inspired robots, Mobile Robot Olfaction/Plume Tracking, Embedded System/Machine vision-based Systems /Virtual reality and Artificial Intelligence. He can be contacted at email: ajm@jxust.edu.cn & moshaydi@gmail.com.







**Zuyan Chen**     graduated from the information engineering college, Jiangxi University of Science and Technology, Currently, he is a member of RARL, and his field of interest is Machine Vision and embedded systems. He can be contacted by email: zuyan\_chen@outlook.com.



**Professor Liao Liefu**     received a B.E. degree in computer science and technology from Central South University, Changsha, China, in 1997, an M.E. degree in automatic control engineering from Jiangxi University of Science and Technology, Ganzhou, China, in 2003, and a PhD degree in system management science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2011. He is currently a dean and graduate tutor at, School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, China, researching areas: artificial intelligence and neural network. He can be contacted at email: liaolf@126.com.



**Dr. Shuai Li**     received a B.E. degree in precision mechanical engineering from Hefei University of Technology, Hefei, China, in 2005, an M.E. degree in automatic control engineering from University of Science and Technology of China, Hefei, in 2008, and PhD degree in electrical and computer engineering from Stevens Institute of Technology, Hoboken, NJ, USA, in 2014. He is currently an Associate Professor (Reader) at Swansea University, Wales, UK, leading the Robotic Lab, conducting research on robot manipulation and impedance control, multi-robot coordination, distributed control, intelligent optimization and control, and legged robots. He can be contacted at email: shuai.li@swansea.ac.uk.